Different architectures for optical packet-switching fabrics have been proposed. Optical cross-connects capable to reconfigure on the nanosecond time scale seem to be the best candidates for a switch core due to their simplicity. Namely, the complexity and cost of the optical technology are very high, so that simplicity of the switch core is essential. Key fast switching optical devices that can be used in packet switches are semiconductor optical amplifiers (SOA) and rapidly tunable lasers.

In the most straightforward design, a packet switch with N inputs and N outputs require N^2 SOAs which are playing the role of gates. However, by combining WDM with space division multiplexing, the overall switch complexity measured in the number of SOAs is significantly reduced: the number of SOAs in a switch is decreased to 2N $\sqrt{N}$ while $\sqrt{N}$ $\sqrt{N}$ x $\sqrt{N}$ waveguide grating routers (WGR) are added. The 256 x 256 switch with the switching time of 5 ns has been demonstrated. If line bitrate is 10 Gbps, short packets of 64 bytes last 64 ns and could be successfully switched in the proposed architecture. The total switching capacity in that case would be 256 x 10Gb/s = 2.56 Tb/s.

Alternatively, each input of a packet switch is equipped with the fast tunable laser which is connected to the outputs through large WGR. Fast tunable laser tunes to the wavelength that will be routed by WGR to the packet designated destination. The 80 x 80 switch with the switching time of 100 ns has been demonstrated. Thus, the proposed architecture would switch only longer packets. But, the long switching time is the result of the driver design and not the laser limitation. It has been shown in that the same laser can tune among wavelengths within less than 15ns.

A. Protocol Description

The WPIM and WRRGS protocols compare similarly as the PIM and RRGS protocols. The PIM protocol consists of several iterations: all inputs send requests to the outputs for which they have packets to send, requested outputs send acknowledgements to their selected inputs, and--.

Please replace Page 8, line 1 through page 11, line 18, with the following:

--i reserves an output for time slot k + N - i within frame $\lceil (k + N - i)/F \rceil$, where $\lceil x \rceil$ is the smallest integer not smaller than x. Also, input i resets its counters cij, $0 \leq j \leq N - 1$, in time slots mF + 1 - N + i, where $m \geq \lceil N/F \rceil$. Time diagram for this first case of WRRGS applied in a 5 x 5 switch is shown in Figure 3. This figure shows the relation between inputs and the time slots for which they are choosing their outputs. For example, in time slot T_5, input I_1 is scheduling or choosing an output for transmission during time slot T_9, while I_3 is scheduling for time slot T_7 and so on. After it chooses an output, e.g., input I_1 forwards the control information (about available outputs) to input I_2 which reserves an output for time slot T_9 in the next time slot T_6. Bold vertical line denotes that input I_o starts a new schedule choosing any of the outputs, i.e. it does not receive the control information from input I_4.

Pipelining proposed for RRGS in might be applied to WRRGS in order to equalize inputs. Time diagram for this case of WRRGS applied in a 5 x 5 switch is shown in Figure 4. Here, in each time slot another input starts a schedule. But, an input might interchangeably reserve outputs for different frames. For example, input I_o reserves an output for time slot T_{11} in time slot T_6, and it reserves an output for time slot T_9 in the next time slot T_7. If the frame length is F = 5, then input I_o interchangeably reserves outputs for frames F_3 and F_2. For a reasonable assumption that $F \geq N$, an input might interchangeably reserve outputs for at most two consecutive frames. So, each queue should be assigned multiple counters related to different frames. Depending on the future time slot for which an input reserves an output, a specified counter of the chosen queue will be decremented by one. Counters are reset every F time slots.

Let us now consider all 1-9 steps of the pipelined WRRGS, including service of the best effort traffic. In any time slot k, each input chooses outputs for two different time slots in future, k + N - i and $k + 2 \cdot N - i$ within frames $\lceil (k + N - i)/F \rceil$ and $\lceil (k + 2 \cdot N - i)/F \rceil$. First, an input reserves an output with the positive counter for time slot $k + 2 \cdot N - i$, then, it reserves any output for time slot k + N - i. Also, input i resets its counters cij, $0 \leq j \leq N - 1$, in time slots $mF + 1 - 2 \cdot N + i$, where $m \geq \lceil 2 \cdot N/F \rceil$. Figure 5 shows the time diagram for all 1-9 steps of WRRGS applied in a 3 x 3 switch. For example, in time slot T_7, input I_1 chooses one of the available prioritized outputs for the time slot T_{12}, and then it chooses any of the available outputs for time slot T_9. This is because input I_1 uses its first chance to schedule for time slot T_{12} in time slot T_7,

and, therefore, it considers only queues with positive counters. On the other side, input I_1 uses the second chance to schedule for time slot T_9 in time slot T_7, and, therefore, it considers all queues for service. It is possible to equalize inputs assuming service of the best-effort traffic as well.

C. Protocol Performance

It is essential to determine the portion of the switch capacity that a scheduling algorithm can share among the inputs. More precisely, we want to determine the maximum admissible utilization, p, of any input or output line:

$$\sum_m p_{im} = \frac{1}{F}\sum_m a_{im} \le p,$$

$$\sum_m p_{mj} = \frac{1}{F}\sum_m a_{mj} \le p,$$

$$0 \le i, j \le N-1,$$

B2

which can be guaranteed to the input-output pairs. So, if input-output pair (i, j) requests a new portion of bandwidth, Δp_{ij}, it is accepted if:

$$\sum_m p_{im} + \Delta p_{ij} \le p,$$

$$\sum_m p_{mj} + \Delta p_{ij} \le p,$$

and input-output pair (i, j) is assigned $\Delta a_{ij} = \lceil \Delta p_{ij} \cdot F \rceil$ new time slots per frame. We will prove that p = 1/2 for the WRRGS, due to the fact that the RRGS finds a maximal matching between inputs and outputs.

Lemma 1: The WRRGS protocol ensures a_{ij} time slots per frame to input-output pair (i, j), $0 \le i, j \le N - 1$, if the following condition holds:

$$\sum_m a_{im} + \sum_m a_{mj} - a_{ij} \le F, \qquad (2)$$

Proof: Only prioritized packets are being viewed as if WRRGS consists only of steps 1-5.

Observe time slots within a frame in which either input i or output j are connected, but they are not connected to each other. In each of these time slots, sum $s_{ij} = \sum_{m \neq j} c_{im} + \sum_{m \neq i} c_{mj}$ is greater than 0, and then it is decremented by at least 1. Sum s_{ij} is the largest at the beginning of a frame and from (2), it fulfills:

$$s_{ij} = \sum_{m \neq j} a_{im} + \sum_{m \neq i} a_{mj} \leq F - a_{ij}, \qquad (3)$$

As a conclusion, in at least a_{ij} time slots per frame neither input i is connected to some output other than j, nor output j is connected to some input other than i. In these time slots, input i reserves output j if there are packets in queue (i, j) and unused credits $c_{ij} > 0$. This is because none of the inputs have chosen output j before input i, and input i is not choosing any other output. Therefore, input i will choose output j as supposed by RRGS, and by any other algorithm that finds a maximal matching between inputs and outputs. In summary, if condition (2) is fulfilled then a_{ij} time slots per frame are guaranteed to input-output pair (i, j).

Lemma 2: The WRRGS protocol ensures a_{ij} time slots per frame to input-output pair (i, j), $0 \leq i, j \leq N - 1$, if the following condition holds:

$$\sum_m a_{im} \leq \frac{F+1}{2},$$

$$\sum_m a_{mj} \leq \frac{F+1}{2}, \qquad (4)$$

Proof: From inequality (4), it follows that:

$$\sum_m a_{im} + \sum_m a_{mj} \leq F+1 \Rightarrow$$

$$\sum_m a_{im} + \sum_m a_{mj} - a_{ij} \leq F,$$

since $a_{ij} \geq 1$. Because inequality (4) implies inequality (2), Lemma 1 directly follows from Lemma 2.

Theorem: The WRRGS protocol ensures p_{ij} of the line bit-rate to input-output pair (i, j), $0 \leq i, j \leq N - 1$, if the following condition holds:

$$\sum_m p_{im} \leq \frac{1}{2},$$

$$\sum_m p_{mj} \leq \frac{1}{2}, \qquad (5)$$

Proof: Condition (5) implies (4), so Theorem follows from Lemma 2.

The above theorem holds for the WPIM as well, considering the fact that PIM finds a maximal matching between inputs and outputs.

B2

Admission control in WRRGS is simple, new Δa_{ij} time slots are assigned to input-output pair (i, j) if:

$$\sum_m a_{im} + \Delta a_{ij} \leq \frac{F+1}{2},$$

$$\sum_m a_{mj} + \Delta a_{ij} \leq \frac{F+1}{2}, \qquad (6)$$

Central controller does not have to precompute schedule when a new request is admitted. Only input i has to update the value of $a_{ij} \leftarrow a_{ij} + \Delta a_{ij}$, $0 \leq j \leq N - 1$, in order to set the correct counter value $c_{ij} = a_{ij}$ at the beginning of each frame. Consequently, WRRGS can follow fast changes of traffic pattern.--.

Please replace Page 13, line 1 through line 30, with the following:

-- information about outputs reserved in each time slot of a frame. An input module also stores a table about its reserved output in each time slot of a frame. Moreover, the time slot duration can be very short in circuit switches, so that a selection takes multiple, e.g. r, time slots to be calculated. It follows that the bandwidth allocation can be changed in every block of r frames.

B3

Both bandwidth reservation and release are based on credits. At the beginning of a block of frames, each counter is loaded to the difference of the number of time slots newly assigned to the input-output pair, and the number of time slots released by this pair.

If the counter value is negative, an input-output pair releases its previously assigned time slot and increments the counter by one, until it becomes zero. Otherwise, if the counter value is positive, an input-output pair reserves time slots in a frame, and decrements the counter until its value is zero. As before, new bandwidth is allocated to some input-output pair if inequalities are fulfilled. Inputs sequentially release previously assigned time slots and then sequentially reserve admitted time slots, one after another. Pipelining can be applied. For example, input i releases an output $r(2N - i + 1)$ time slots in advance, and reserves an output $r(N - i + 1)$ time slots in advance. Input picks up output that has not been reserved in some of the previous blocks of frames, or by some of the previous inputs which reserve the outputs for the same time slot in the current block of frames. Note that each node would learn about all released and reserved outputs for some future time slot exactly $r \cdot N$ time slots after it releases or reserves an output for that time slot. So, the node can store the information in its tables before the next block of frames as long as $rN \leq rF$, which is the case of interest. In a conclusion, in arbitrary block of frames the scheduler accepts new bandwidth requests, in the next block of frames it calculates new time slot assignment, and finally in the third block of frames the circuits are switched according to the new schedule. Of course, this process is also pipelined so that the switch time slot assignment can be changed at the beginning of each block of frames.

In accordance with the present invention, a simple way to flexibility share bandwidth in switches with input buffering has been described. The simplicity of the proposed protocol makes it attractive for switching of several Tb/s, assuming the current technology. It has also been shown that the proposed WRRGS can share at least 50% of the total switch capacity.

WRRGS has several desirable features. First, WRRGS algorithm can serve traffic with fast varying bandwidth requirements typical in data networks. Second, WRRGS requires simple--.